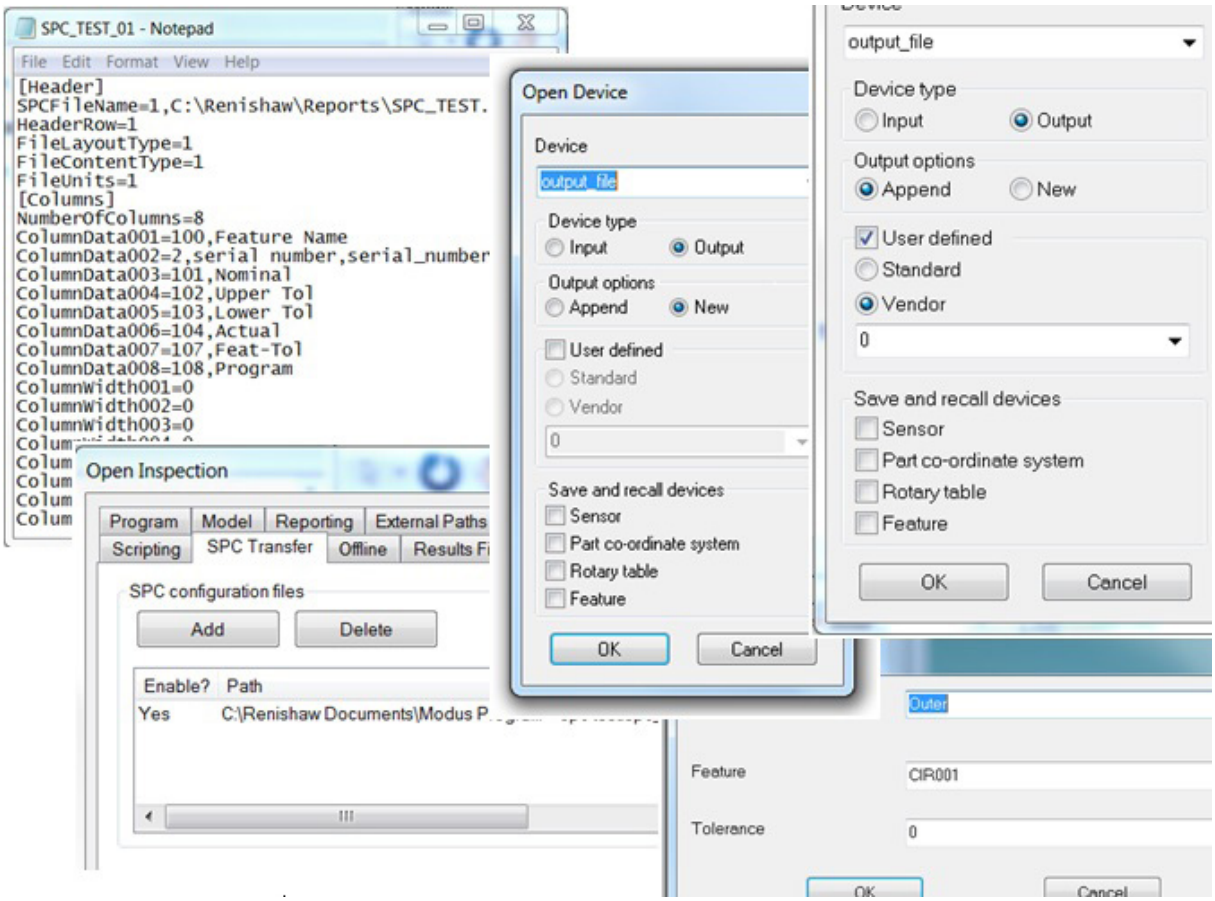
Tutorial
H-1000-5337-01-C

RENISHAW
apply innovation™

Alternative user defined outputs



Renishaw part no: H-1000-5337-01-C

Issued: 08 2014

Alternative user defined outputs

Care of equipment

Renishaw probes and associated systems are precision tools used for obtaining precise measurements and must therefore be treated with care.

Changes to Renishaw products

Renishaw reserves the right to improve, change or modify its hardware or software without incurring any obligations to make changes to Renishaw equipment previously sold.

Warranty

Renishaw plc warrants its equipment for a limited period (as set out in our Standard Terms and Conditions of Sale) provided that it is installed exactly as defined in associated Renishaw documentation.

Prior consent must be obtained from Renishaw if non-Renishaw equipment (e.g. interfaces and/or cabling) is to be used or substituted. Failure to comply with this will invalidate the Renishaw warranty.

Claims under warranty must be made from authorised service centres only, which may be advised by the supplier or distributor.

Trademarks

Windows 98, Windows XP, Windows 2000 and Windows NT are registered tradenames of the Microsoft Corporation.

IBM is the tradename of the International Business Machines Inc

All trademarks and tradenames are acknowledged.

Contents

1 Alternative user defined outputs

1.1 Tutorial pre-requisites

- The student should understand the contents of the basic MODUS tutorials
- The student should have completed the ‘Introduction to high level programming’ tutorial
- The student should have completed the ‘Curve manipulation - trimming’ tutorial

1.2 Tutorial objectives

- Introduction to the automated management of output files from a measurement program
- Introduction to output methods that permit compatibility with third-party data processing software

2 Introduction

This tutorial covers the most common techniques used to automatically manage the output files produced by MODUS. This includes methods of automatically archiving output files and creating records of results from previously-run programs.

Also, methods of customising the format of outputted data are discussed. This allows measurement data to be used in third-party analysis software packages (SPC software for instance).

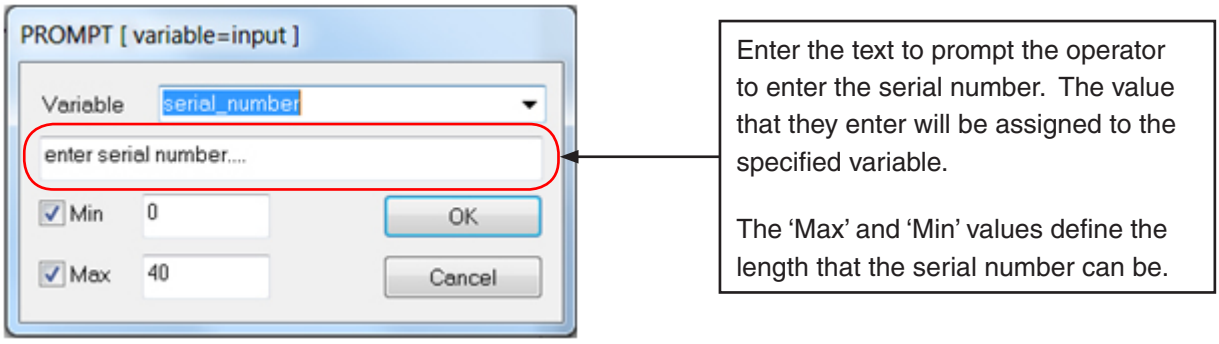
To aid with training continuity it is suggested that the program written during the ‘Curve manipulation - trimming’ tutorial should be used during this tutorial.

3 Creating a custom defined result file

A custom defined path and file name can be created in a part program by using a prompt and variables. After the user fills in the prompt information, a file will be created in a specified location. In this example, a path and serial number will be combined in a variable to specify the file name and location.

Following on from the ‘Curve manipulation - trimming’ tutorial, declare a string variable to store the serial number of a part to be measured (e.g.serial_number).

Use the high level command ‘Simple Prompt’ to ask for the serial number to be input from the operator.

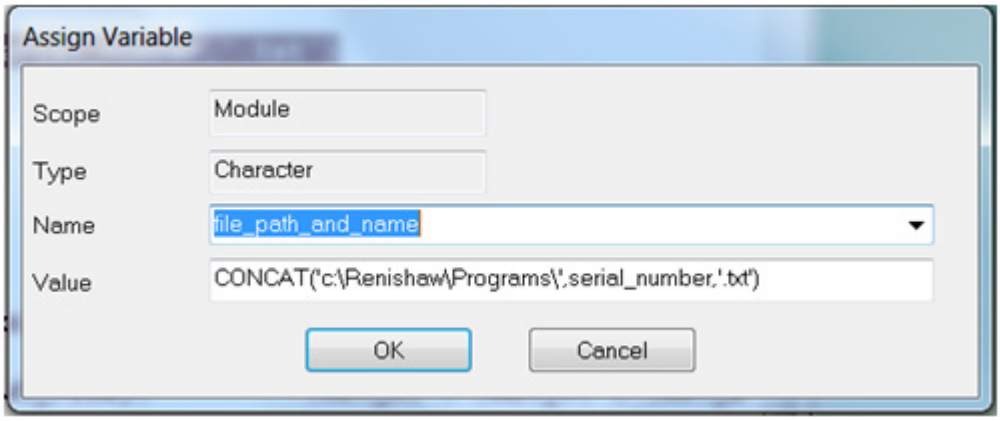


Example code:

```
DECL/LOCAL,CHAR,512,serial_number
serial_number=PROMPT/'Enter serial number',40,0
```

The path, serial number and file extension can be combined using the CONCAT statement.

Declare a string variable that will be used to hold the file and path information (e.g. file_path_and_name).



‘Name’ is the variable declared previously and ‘Value’ is the information to be assigned to that variable. In this example, the path information is combined with the serial number variable and the file extension (e.g. ‘.txt’).

Be very careful that the syntax in the 'Value' line is correct. A missing apostrophe or comma will produce an error and will need to be corrected before MODUS will allow the code to be executed.

The following is an example of the information in the assignment statement that will define the location and file name where output data will be saved.

Example code:

file_path_and_name=ASSIGN/CONCAT('c:\Renishaw\Programs\',serial_number,'.txt')

This is the path where the file will be stored. It can be altered to whatever destination is required. It is enclosed by apostrophes.

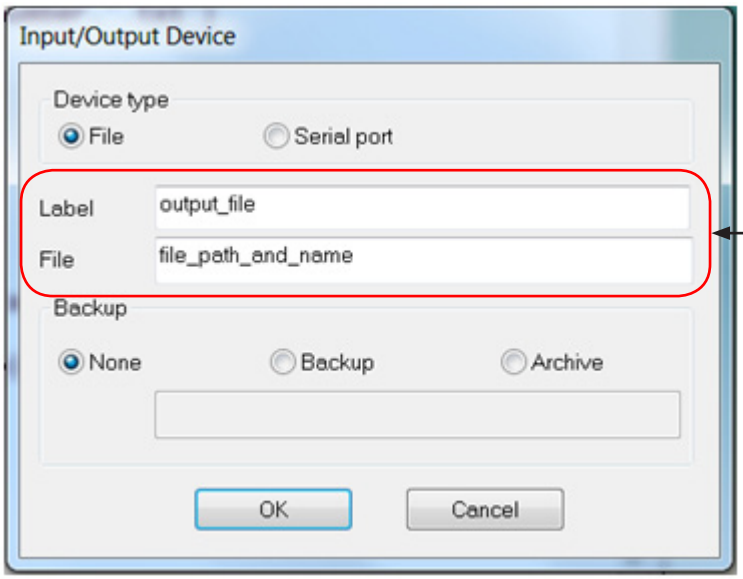
This is the variable that will give the file its name. It is NOT enclosed by apostrophes.

This is the file extension, which gives the results file its format and is enclosed by apostrophes.

The variable file_path_and_name has the pathway, name and extension combined as a single variable. This will be used in a device command to specify the location of output data.

GUIDANCE NOTE: Be sure to separate each piece of information with a comma.

Select 'High Level' and 'Device'. This creates a link to the file, replacing the file name and path with a meaningful label. The previous variable is placed in the 'File' text box. The file will now be referred to with the label rather than the file and path.



In the 'Label' box write a label for this input / output device. Use any meaningful label name (e.g. output_file).

In 'File', use the path and file name variable.

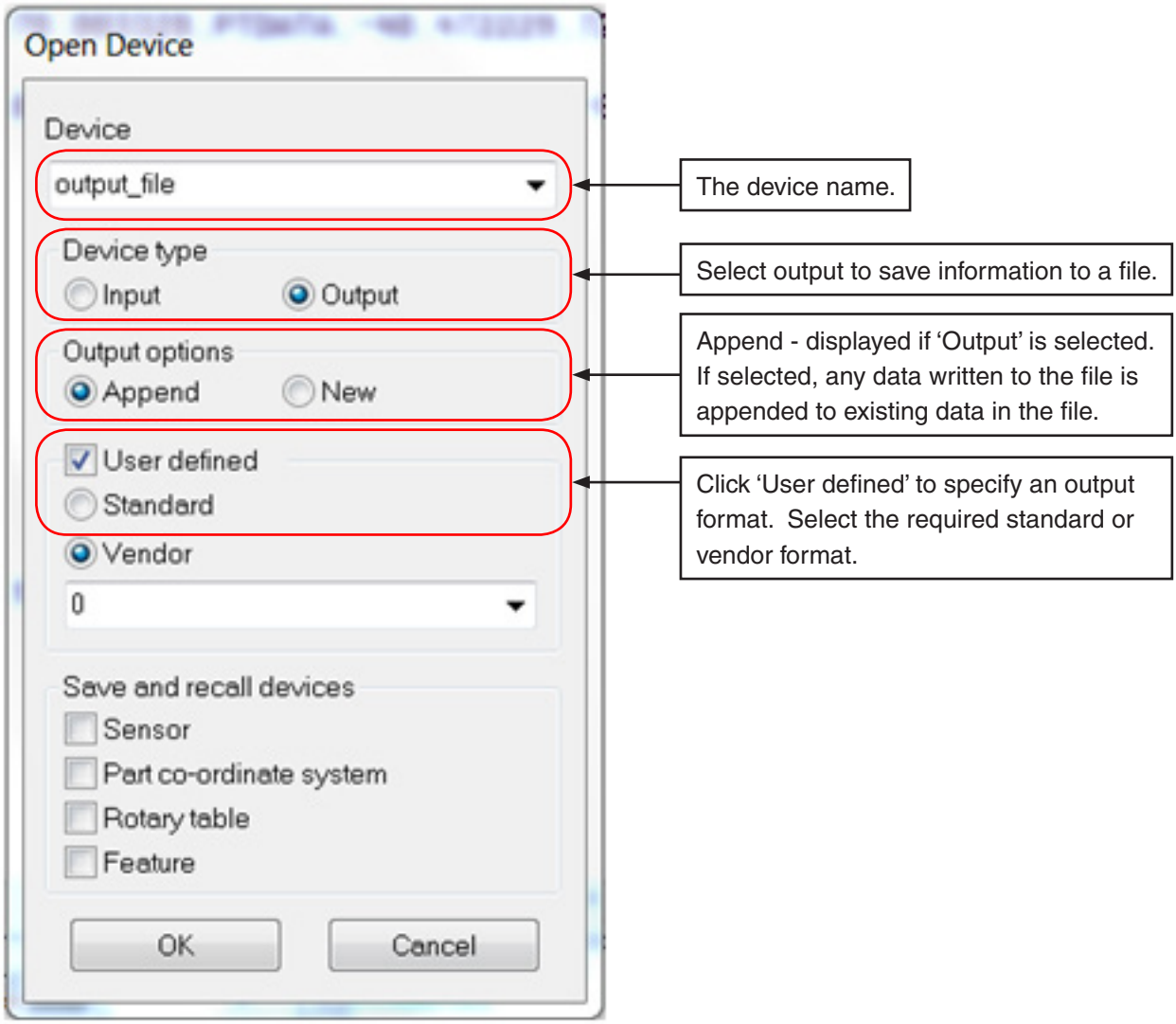
CLicking ‘OK’ produces the following code:

DID(output_file)=DEVICE/STOR,‘file_path_and_name’

The inverted commas need to be removed becuase file_path_and_name is a variable.

Example code:

DID(output_file)=DEVICE/STOR,file_path_and_name



Opening the device file allows data to be written to the specified device file.

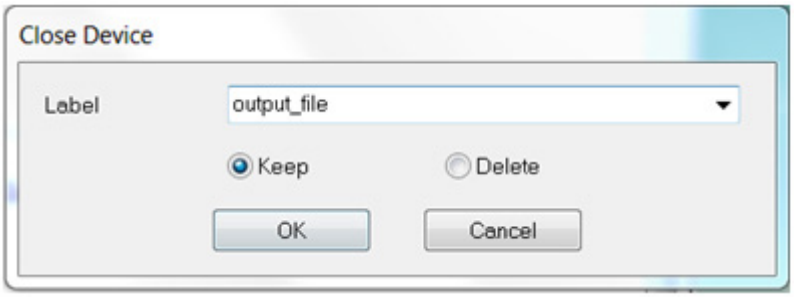
Output the tolerance data for the measured features in the program. This information will appear in the device file.

GUIDANCE NOTE: Be sure that the output data is placed after the DEVICE is opened.

Example code:

```
T(2)=TOL/RAD,-0.1,0.1
T(3)=TOL/POS,2D,0.1,RFS
OUTPUT/FA(ARC001),TA(2),TA(3)
T(4)=TOL/PROFP,-0.1,0.1
OUTPUT/FA(PNT001),TA(4)
OUTPUT/FA(PNT002),TA(4)
OUTPUT/FA(PNT003),TA(4)
```

Use the ‘Close’ command (from high level) to close the file. ‘Keep’ saves the file and ‘Delete’ gets rid of the file when the device is closed. Make sure the ‘Keep radio button is selected.



Example code:

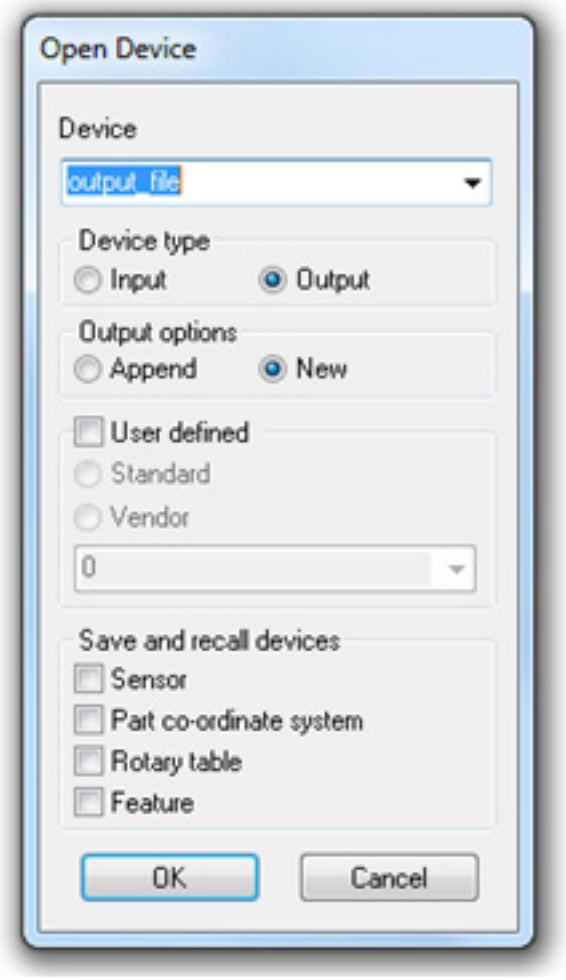
```
CLOSE/DID(output_file),KEEP
```

4 Formatted output of curve data

This section illustrates one way to format output data from a feature. Measured features are sometimes output in a customized format so they can be read to another system. It is common to output GCURVE data in this way to a device file.

Measure a GCURVE, which will be output in a customised format.

Create and open a device in MODUS:

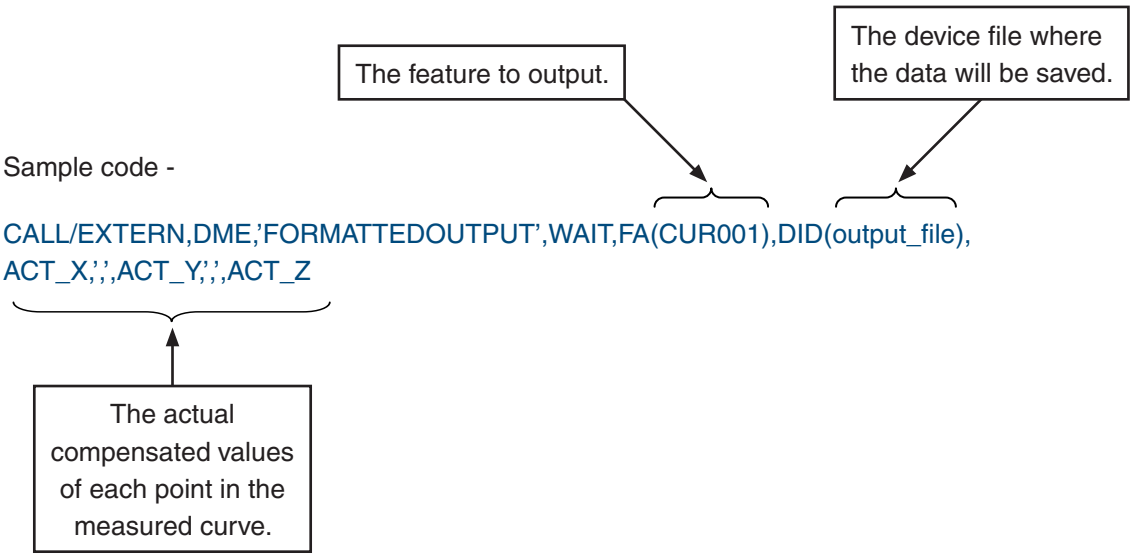


GUIDANCE NOTE: Be sure to uncheck ‘User defined’ which will allow direct output.

Example code:

```
DID(output_file)=DEVICE/STOR,'c:\Renishaw\Programs\123456.txt'
OPEN/DID(output_file),DIRECT,OUTPUT,OVERWR
```

Creating a formatted output requires hand typing, or copying and pasting code, since there is no menu item available from MODUS for a formatted output. The following is a sample of code that can be used when typing in similar code. It should be on a single line in MODUS, but is on separate lines in this tutorial due to space limitations



The sample code produces the following output on a sample curve (first three points only):

```
59.500,     36.470,     -2.000,
59.028,     33.542,     -2.000,
57.675,     30.902,     -2.000,
```

There are many ways to format data. The MODUS help menu has many examples of other parameters that can be used to output actual, nominal, path, raw and other types of data. Generally most people require actual surface point nominals for third party software. See 'Outputting formatted feature data to a device' for comprehensive help on this subject.

GUIDANCE NOTE: It can be seen that there are numerous commas required in the code line, which makes checking the code difficult and the possibility of errors great. This issue could be addressed, and make checking easier, if commas that appear between inverted commas were made a variable. Using this convention each comma between inverted commas would be replaced by the word 'comma'. Below is an example of the code required to do this.

```
DECL/LOCAL,CHAR,512,Comma
Comma=ASSIGN/','
```

e.g. ACT_X,Comma,ACT_Y,Comma,ACT_Z

5 SPC formatted outputs

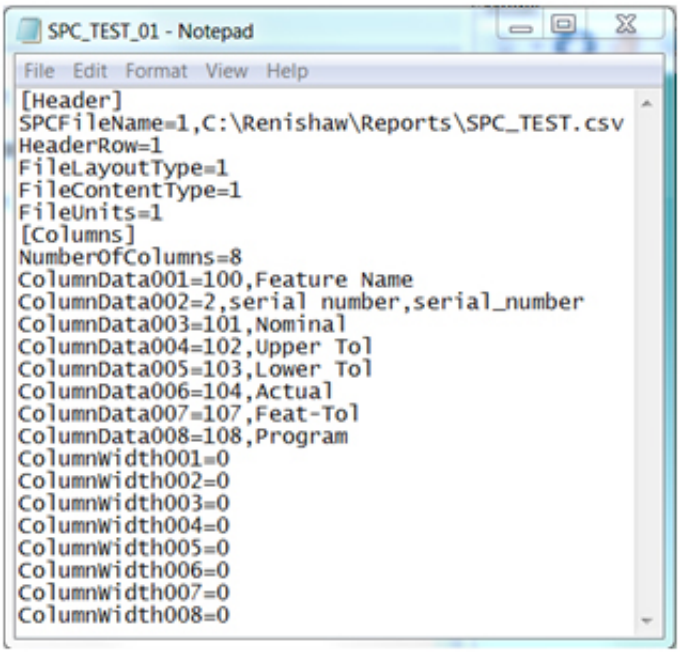
Many third party SPC packages require formatting of data in a specific way. This section of the tutorial shows an example of how to do this.

5.1 SPC transfer configuration file format

MODUS uses a configuration file to define the format of output data. There are two main sections:

1. **Header section -** The header section defines some parameters on how the file is created. For example, it is possible to specify how the name is created:
    a. Hardcoded in the file (SPCFileName=1)
    b. Defined by a variable in the DMIS file (SPCFileName=2)
    c. Keeps the same name as the DMIS program (SPCFileName=3)
2. **Columns section -** This defines which data is output and the text that will appear at the top of each column.

Here is a simple example of a configuration file:



The column data in the example file is as follows:

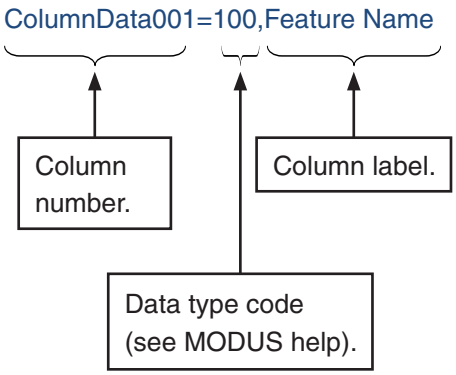
First column - This will output the feature name of what is being measured.

Second column - This will output the contents of the variable serial_number (this could be any variable choosen, give the variable a header name first, in this case serial number (no underscore) before inputting the variable serial_number).

Third to eighth column - Outputs the specified data exactly as described, with ‘Program’ being the name of the overall program run.

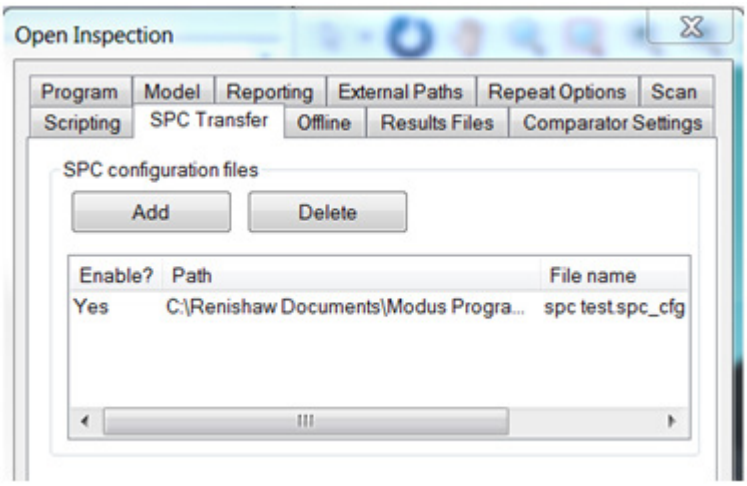
This shows how each column in the configuration file gets formatted.

Example:



The file extension for the configuration file can be .spc_cfg or any other extension specified. For a detailed example on how to build a configuration file in the INI format please refer to the MODUS help document entitled ‘SPC Transfer Configuration File Format’.

Once the SPC configuration file is created, it must be linked to a MODUS program:



On opening a new inspection, select the ‘SPC Transfer’ tab and click ‘Add’. Then, navigate to the configuration file and select it. Finally, make sure the ‘Enable property’ is set to ‘Yes’.

6 Key characteristics

After measuring features that need to be output to a user defined format, key characteristics can be defined. A key characteristic is defined so the output of a tolerance is named in a specific way when it is sent to the SPC transfer file.

In the example that follows, the feature CIR001 has two tolerances that will be used to output circle feature information:

1. T(TRUE_POSITION)

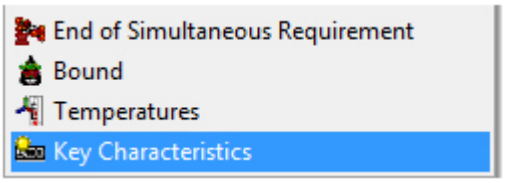
2. T(DIAM_1)

By using a key characteristic, these can be associated with a different name that is more meaningful. Using key characteristic associations:

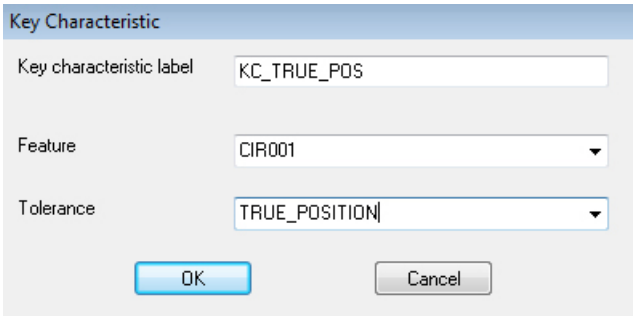
1. T(TRUE_POSITION) will be output to the csv file as KC_TRUE_POS

2. T(DIAM_1) will output to the csv file as KC_DIAM

To create a key characteristic association, select ‘Output’ and choose ‘Key Characteristics’.



The key characteristic prompt will open. Fill in a meaningful label and select the feature and tolerance to associate with the key characteristic.



Example code:

KC(KC_TRUE_POS)=KEYCHAR/F(CIR001), T(TRUE_POSITION)

Repeat the above step for T(DIAM_1).

Finally, create an output for the tolerance of the feature.

Example code:

OUTPUT/FA(CIR001),TA(TRUE_POSITION),TA(DIAM_1)

This will then populate the .csv file with the required fields from the configuration file.

GUIDANCE NOTE: The KC command MUST come before the output command or the key characteristic information will not be sent to the .csv file.

These are the resulting dmis & .csv file.

```
000020  F(CIR001)=FEAT/CIRCLE,INNER,CART,0,0,-10,0,0,1,56
000021  MEAS/CIRCLE,F(CIR001),5
000022  ENDMES
000023  F(CIR002)=FEAT/CIRCLE,INNER,CART,24.395,-24.395,-10,0,0,1,7
000024  MEAS/CIRCLE,F(CIR002),5
000025  ENDMES
000026
000027  T(DIAM_1)=TOL/DIAM,-0.1,0.1
000028  T(TRUE_POSITION)=TOL/POS,2D,0.1,RFS
000029
000030  KC(KC_TRUE_POS)=KEYCHAR/F(CIR001),T(TRUE_POSITION)
000031  KC(KC_DIAM)=KEYCHAR/F(CIR001),T(DIAM_1)
000032
000033  KC(KC_TRUE_POS_2)=KEYCHAR/F(CIR002),T(TRUE_POSITION)
000034  KC(KC_DIAM_2)=KEYCHAR/F(CIR002),T(DIAM_1)
000035
000036  OUTPUT/FA(CIR001),TA(DIAM_1),TA(TRUE_POSITION)
000037  OUTPUT/FA(CIR002),TA(DIAM_1),TA(TRUE_POSITION)
```

	A	B	C	D	E	F	G
1	Feature Name	Nominal	Upper Tolerance	Lower Tolerance	Actual	Feat-Tol	program
2	KC_DIAM	56	0.1	-0.1	56.009	CIR001_DIAM_1	SPC
3	KC_TRUE_POS	0	0.1	0	0.027	CIR001_TRUE_POSITION	SPC
4	KC_DIAM_2	7	0.1	-0.1	7.006	CIR002_DIAM_1	SPC
5	KC_TRUE_POS_2	0	0.1	0	0.023	CIR002_TRUE_POSITION	SPC

Renishaw plc

New Mills, Wotton-under-Edge,
Gloucestershire, GL12 8JR
United Kingdom

T +44 (0)1453 524524
F +44 (0)1453 524901
E uk@renishaw.com

www.renishaw.com

RENISHAW
apply innovation™

For worldwide contact details, please visit our main web site at www.renishaw.com/contact

H-1000-5337-01